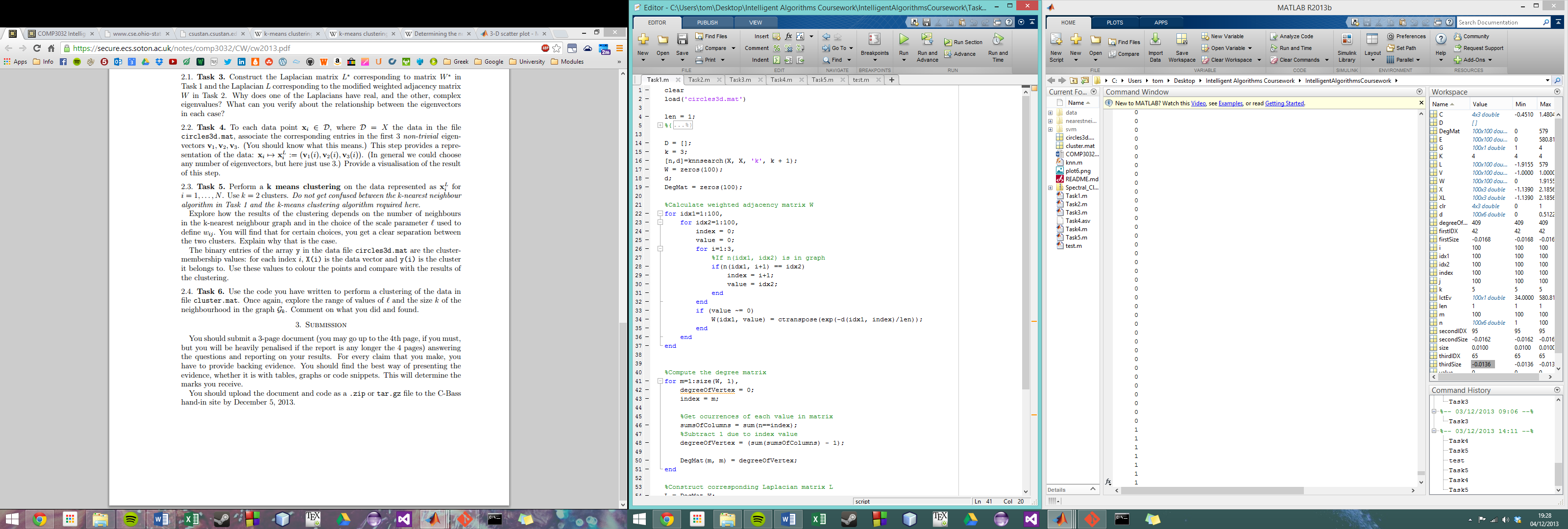
COMP3032 COURSEWORK

DATA REPRESENTATION AND CLUSTERING USING GRAPH LAPLACIANS

# Section 1: Representing data using nearest-neighbour graphs

# Task 1

I’m using Matlab to implement the required matrix operations for its power and versatility. I used Matlab’s knnsearch() function to find the *k-nearest neighbours*of each vertex of the data and the distance to each vertex. The mathematics functions allowed me to produce the weighted adjacency matrix. I summed the occurrences of each index in the KNN graph to populate the degree matrix and a conjugate transpose function to give the argument for the eig() function, giving the eigenvalues of the Laplacian.



Figure

## Questions

### Why are the Eigenvalues not all real?

The Eigenvalues are not all real, because the Laplacian is non-symmetric. The entries of the Laplacian are all read-valued, but this does not mean its eigenvalues will all zero imaginary parts.

The Laplacian matrix could have been normalised using a symmetric normalisation algorithm. The Laplacian is not symmetric and so its eigenvalues are not as a matter of necessity, all real-valued.

Alternatively a random walk normalised Laplacian could have been used. This would produce a matrix similar to a scalar perturbation of the symmetric normalised Laplacian and as such has real eigenvalues.

# Task 2

I shared the weight of connected nodes and in the process ensured the adjacency matrix was symmetric, by applying a weight (connecting the vertices) only if i was among the k-nearest neighbours of j and j was among the k-nearest neighbours of i. This resulted in a symmetric adjacency matrix as seen graphically when Task2.m is executed.

# Section 2: Embedding the data and clustering using the eigenvectors of the Laplacian

# Task 3

I combined my code to show the conjugate transpose of the non-symmetric Laplacian matrix from Task 1 and the symmetric Laplacian from Task 2. When I plotted the two matrices using the spy() function in Matlab, I observed a symmetric image about the diagonal for the symmetric Laplacian as expected. The fact that the second adjacency matrix is not a conjugate transpose does not affect its value since it is symmetric and has real-valued entries.

## Questions

### Why does one of the Laplacians have real, and the other, complex eigenvalues?

The first Laplacian has complex eigenvalues as in Task 1, since it is an indirect mapping and consequently non-symmetric. The second Laplacian has all real-valued eigenvalues since the weighting equation shares weights in both directions and thus guarantee’s symmetry. Since it is real valued and symmetric, it is equal to its conjugate transpose and is a self-adjoint matrix (Hermitian). Every Hermitian matrix has real-valued eigenvalues.

### What can you verify about the relationship between the eigenvectors in each case?

An eigenvector is a non-zero vector that, when multiplied by the square matrix (in our case Laplacian matrix), yields a constant multiple of itself, where the multiplier is the eigenvalue. If every eigenvalue in our case is real, then the eigenvectors will be real. If there are complex eigenvalues as for the second Laplacian, there will be complex eigenvectors also. This was verified in Matlab in Task3.m.

# Task 4

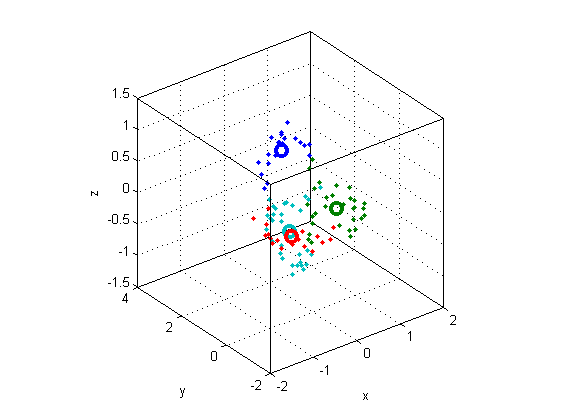
I computed 3 non-trivial eigenvectors with the smallest length, and used them for representing my data. I observed the 3D scatter plot as in Figure 1 with k = 12 and l = 1. My visualisation of the three non-trivial eigenvectors was produced by the scatter3() function in Matlab and demonstrates a good representation of the data.

Figure

# Task 5

I implemented a k-means clustering algorithm to cluster data according to how close it is to one of k centre points. I clustered the original data points X and also the subset of XL. When k was altered the centre points of each cluster changed to be located in areas, generally of higher node density, but sometimes in areas simply where they were the nearest point to the same number of nodes as each of the other clusters (this is the result of k-means clustering).

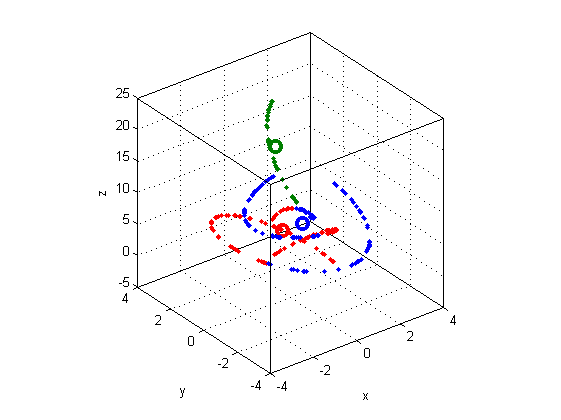
As L was increased from 0.1 through to around 50 there were no significant changes from 10 onwards. From L = 5 to 8, there was some significant changing of the adjacency matrix and consequently the Laplacian matrix, causing the first 3 eigenvectors to change. Figure 2 shows the most natural clustering for the data X, when K = 4.



Figure

# Task 6

I collected the data from cluster.mat and fed it into my k-means clustering algorithm. I used a range of values for L and K, each producing different results. The values of k changed the numbers of clusters and most naturally were clustered into 3 as shown in the graph in Figure 4.



Figure